## CLAIMS

What is claimed is:

1. A method of processing instructions in a microprocessor, comprising:

(a) fetching instructions from an instruction memory, certain fetched instructions being load instructions (loads) and causing load operations, and other fetched instructions being store instructions (stores) and causing store operations;

(b) executing the fetched instructions out of program order;

(c) detecting a load/store order violation wherein a load executes prior to a store on whose data the load depends;

(d) creating a store set for the load;

(e) adding the store to the store set;

(f) determining whether the store is poisoned by a previously poisoned instruction;

(g) if the store is poisoned, setting a poison value that indicates that the store is poisoned; and

(h) re-processing said load if said poison value associated with said store indicates the store has been poisoned.

2. The method of claim 1 wherein (g) includes setting a bit in a table.

3. The method of claim 1 wherein the store set includes a pointer that points to the poison value.

4. The method of claim 1 wherein said store set includes a pair of tables which are used to identify said store instruction.

5. The method of claim 4 further including clearing said poison value when said store is no longer poisoned.

6. A method of processing a store instruction (store) to execute before a load instruction (load) that target a common memory location, comprising:

(a) determining if the data to be written by said store is stale;

(b) if said data is stale, setting a value associated with said store; and

(c) if said value is set, re-processing said load to execute after said data is no longer stale.

7. The method of claim 6 further including establishing a store set for said load to include said store.

8. The method of claim 7 further including using said store set to access said value.

9. The method of claim 6 wherein said value comprises a poison bit.

10. A computer system, comprising:

a microprocessor;

an input device coupled to said microprocessor; and

memory coupled to said microprocessor, said memory containing executable instructions;

wherein said microprocessor:

fetches instructions from said memory, certain fetched instructions being load instructions (loads) and causing load operations, and other fetched instructions being store instructions (stores) and causing store operations;

executes the fetched instructions out of program order;

detects a load/store order violation wherein a load executes prior to a store on whose data the load depends;

creates a store set for the load;

adds the store to the store set;

determines whether the store is poisoned by a previously poisoned instruction;

if the store is poisoned, sets a poison value that indicates that the store is poisoned; and

re-processes said load if said poison value associated with said store indicates the store has been poisoned.

11. The system of claim 10 wherein said poison value comprises a bit in a table.

12. The system of claim 10 wherein the store set includes a pointer that points to the poison value.

13. The system of claim 10 wherein said store set includes a pair of tables which are used to identify said store instruction.

14. The method of claim 13 wherein said microprocessor clears said poison value when said store is no longer poisoned.

15. A computer system, comprising:

a microprocessor; and

memory coupled to said microprocessor, said memory containing a store instruction (store) and a load instruction (load) that target a common memory location;

wherein said microprocessor:

fetches said load and store;

determines if the data to be written by said store is stale;

if said data is stale, sets a value associated with said store;

if said value is set, re-processes said load to execute after said data is no longer stale.

16. The system of claim 15 wherein said microprocessor establishes a store set for said load to include said store.

17. The system of claim 16 wherein said microprocessor uses said store set to access said value.

18. The system of claim 15 wherein said value comprises a poison bit.

19. A microprocessor, comprising:

a fetch stage which fetches executable instructions from memory, certain fetched instructions being load instructions (loads) and causing load operations, and other fetched instructions being store instructions (stores) and causing store operations;

an execution stage coupled to said fetch stage which executes the fetched instructions out of program order; and

logic coupled to said fetch and execution stages that detects a load/store order violation wherein a load executes prior to a store on whose data the load depends, creates a store set for the load, adds the store to the store set, determines whether the store is poisoned by a previously poisoned instruction, if the store is poisoned, sets a poison value that indicates that the store is poisoned, and re-processes said load if said poison value associated with said store indicates the store has been poisoned.

20. The microprocessor of claim 19 wherein said poison value comprises a bit in a table.

21. The microprocessor of claim 19 wherein the store set includes a pointer that points to the poison value.

22. The microprocessor of claim 19 wherein said store set includes a pair of tables which are used to identify said store instruction.

23. The microprocessor of claim 22 wherein said logic clears said poison value when said store is no longer poisoned.

24. A microprocessor, comprising:

a fetch stage which fetches instructions including a store instruction (store) and a load instruction (load) that target a common memory location; and

logic coupled to said fetch stage which determines if the data to be written by said store is stale, and if said data is stale, sets a value associated with said store and re-processes said load to execute after said data is no longer stale.

25. The microprocessor of claim 24 wherein said logic establishes a store set for said load to include said store.

26. The microprocessor of claim 25 wherein said logic uses said store set to access said value.

27. The microprocessor of claim 24 wherein said value comprises a poison bit.